MLCC_SXC

MACHINE LEARNING
CRASH COURSE

FEATURE CROSSES AND REGULARIZATION

Week 2

- Jimut Bahan Pal
jimutbahanpal@yahoo.com

TRAINING AND TEST SET

What are training and test set ?

TRAINING AND TEST SET

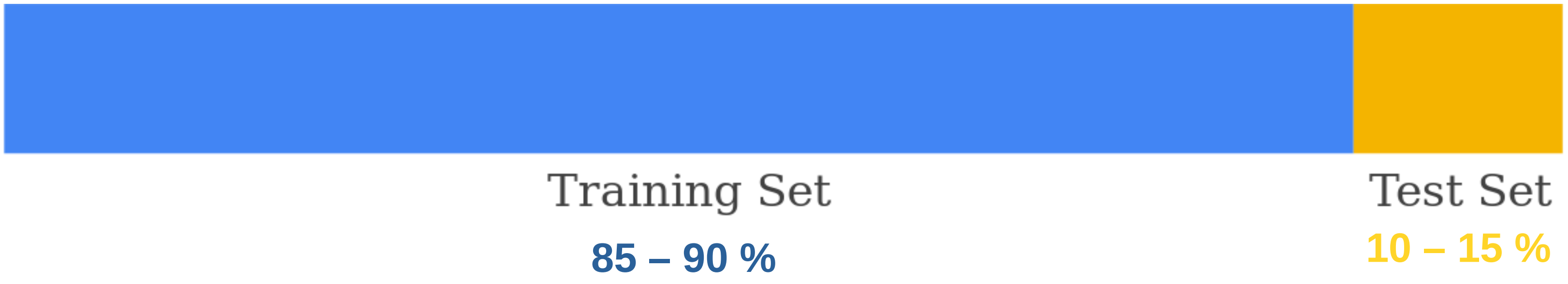
What are training and test set ?

If we have one data set we divide it into two parts, one for training and the other for testing. Now if the data set is big then there is no problem, since about 85 – 90 % would be used for training and the rest 15 - 10 % will be used for testing how well the model predicts!

TRAINING AND TEST SET

What are training and test set ?

If we have one data set we divide it into two parts, one for training and the other for testing. Now if the data set is big then there is no problem, since about 85 – 90 % would be used for training and the rest 15 - 10 % will be used for testing how well the model predicts!



jimutbahanpal@yahoo.com

TRAINING AND TEST SET

Some points to keep in mind while training the model

- The larger the training set, the better the model

jimutbahanpal@yahoo.com

TRAINING AND TEST SET

Some points to keep in mind while training the model

- The larger the training set, the better the model

- The larger the test set, the larger the confidence in the evaluation matrix

jimutbahanpal@yahoo.com

TRAINING AND TEST SET

Some points to keep in mind while training the model

- The larger the training set, the better the model

- The larger the test set, the larger the confidence in the evaluation matrix

- DONOT TRAIN ON THE TEST DATA

    - If you find surprisingly low losses make sure to check if you have trained on the test data or have forgot to randomize the data.

jimutbahanpal@yahoo.com

TRAINING AND TEST SET

Some points to keep in mind while training the model

- It may happen that you are training on the dataset which follows the same pattern, like training the whole dataset on winter values and validating on summer values, this might result in errors, make sure to avoid such errors so that the model predicts well for all values.

jimutbahanpal@yahoo.com

TRAINING AND TEST SET

Some points to keep in mind while training the model

- It may happen that you are training on the dataset which follows the same pattern, like training the whole dataset on winter values and validating on summer values, this might result in errors, make sure to avoid such errors so that the model predicts well for all values.

- We should make the training set such that it represents the data set as a whole.

jimutbahanpal@yahoo.com

TRAINING AND TEST SET

Some points to keep in mind while training the model

- It may happen that you are training on the dataset which follows the same pattern, like training the whole dataset on winter values and validating on summer values, this might result in errors, make sure to avoid such errors so that the model predicts well for all values.

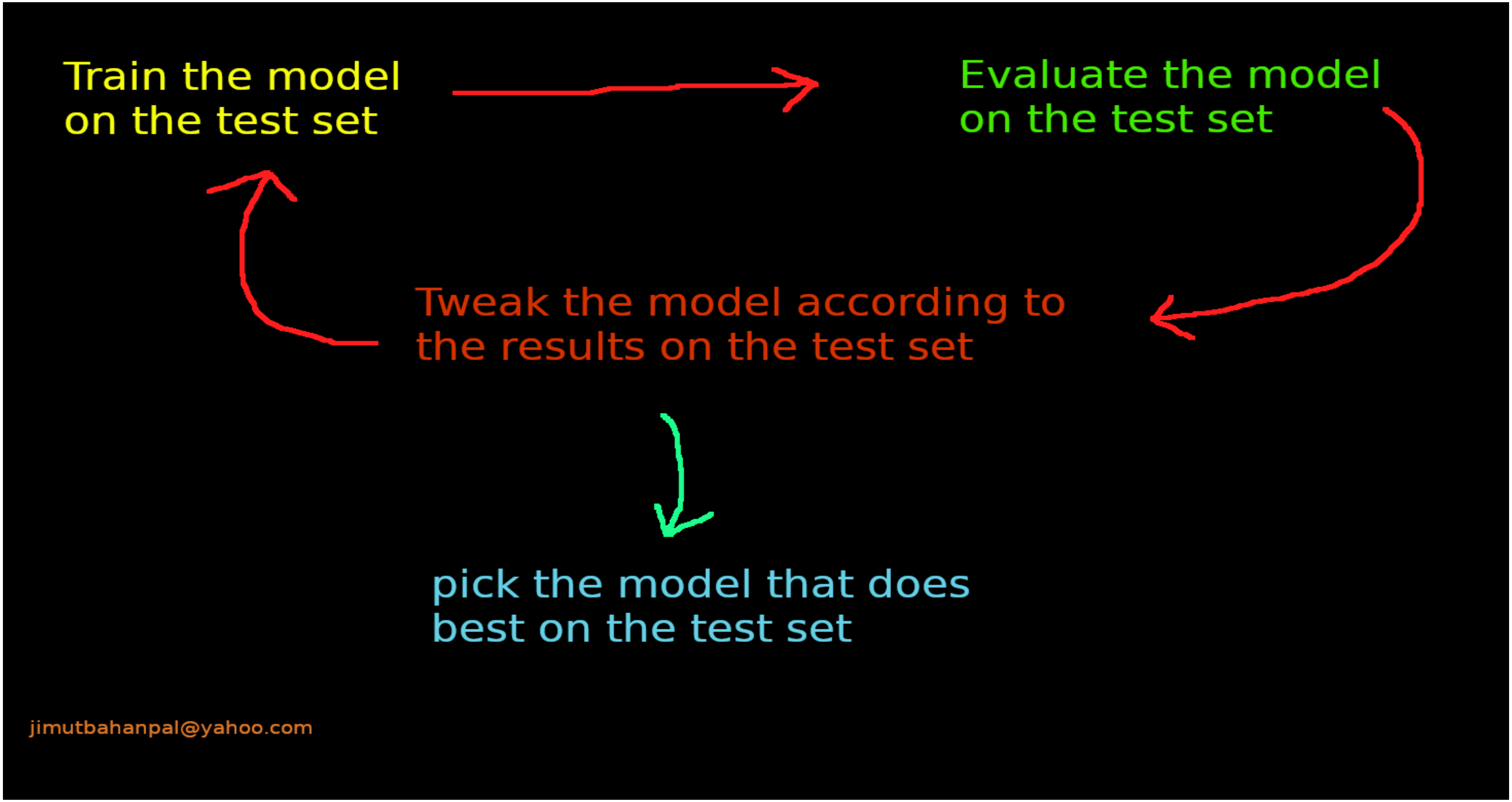
- We should make the training set such that it represents the data set as a whole.

- Test set is large enough to yeild statistically meaningful result.

jimutbahanpal@yahoo.com

TRAINING AND TEST SET

WORK FLOW



TRAINING AND TEST SET

WORK FLOW - The iterative method

We have trained the model on the training data and now we will test the model on the test data and observe the matrix. We will tweak some settings, maybe the learning rate and try again. We will see if we could try to improve the test set accuracy.

jimutbahanpal@yahoo.com

TRAINING AND TEST SET

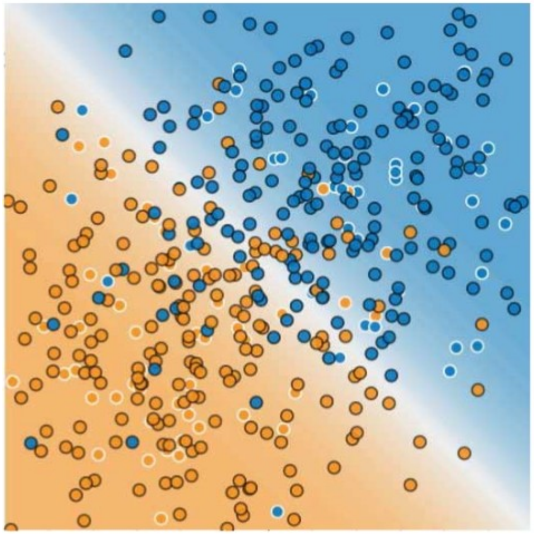
WORK FLOW - The iterative method

We have trained the model on the training data and now we will test the model on the test data and observe the matrix. We will tweak some settings, maybe the learning rate and try again. We will see if we could try to improve the test set accuracy.

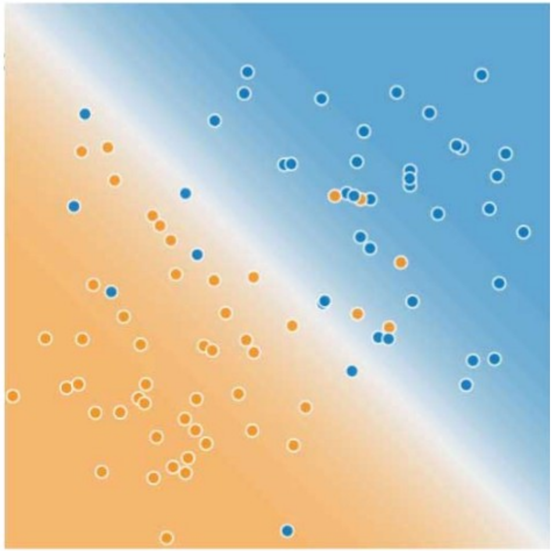
Maybe add some more features in them, maybe take some features out and keep iterating till we find the best iterative model that can predict best in the test set matrix.

TRAINING AND TEST SET

Example of a classification problem



Training Data

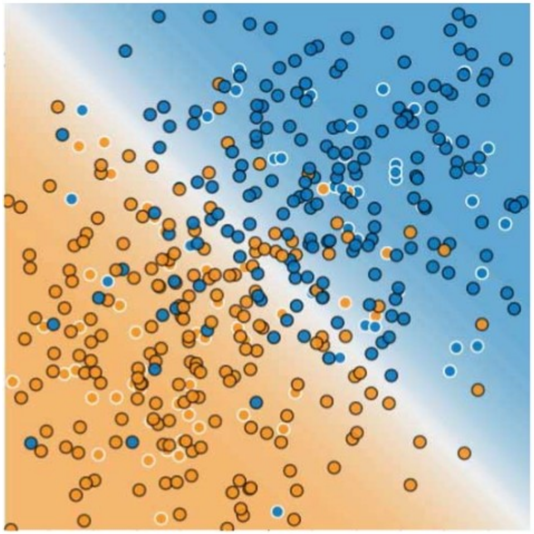


Test Data

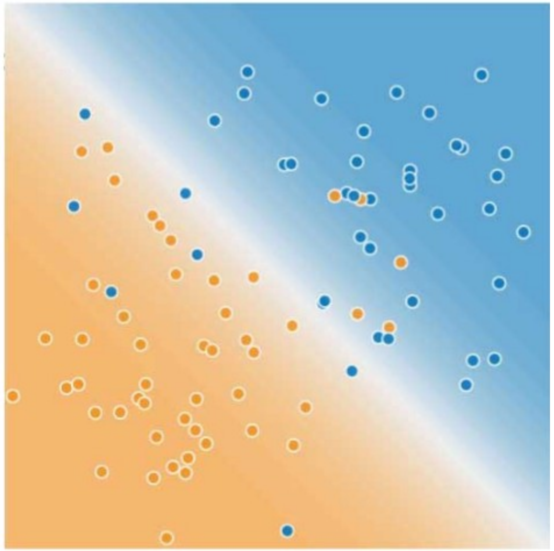
jimutbahanpal@yahoo.com

TRAINING AND TEST SET

Example of a classification problem



Training Data



Test Data

From this example we see that the model is not perfect, but it works well on the test data. This model may result in wrong predictions for some exceptional values but it's doing perfectly fine since the training set also have a few erroroneous exceptional data, and it is not overfitting the data!

jimutbahanpal@yahoo.com

VALIDATION

We now have trained the model in a given set of data and predict for a test set of data.

jimutbahanpal@yahoo.com

VALIDATION

We now have trained the model in a given set of data and predict for a test set of data.

ARE THERE SOME PROBLEMS ?

VALIDATION

Well, we may try to overfit on the peculiarities of the test data...

VALIDATION

Well, we may try to overfit on the peculiarities of the test data...

that’s too is bad

But we can do something :D

Validation

We will partition the training set into another part known as validation set, which results to reduce any case of overfitting by any chance. Use validations to evaluate results from the training set. Use the test set to double check your evaluation after the model has passed the validation set

VALIDATION

Validation

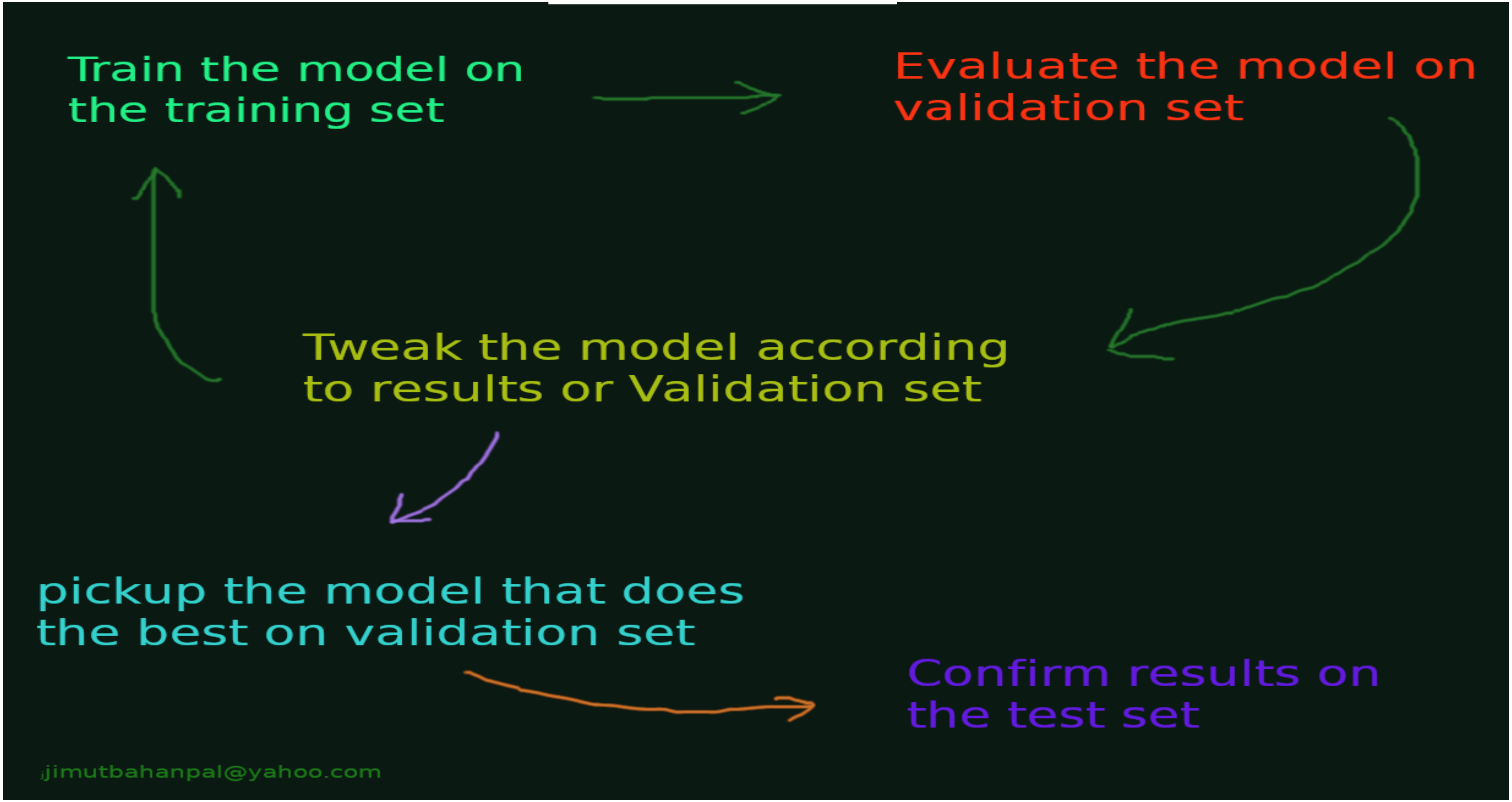
We will partition the training set into another part known as validation set, which results to reduce any case of overfitting by any chance. Use validations to evaluate results from the training set. Use the test set to double check your evaluation after the model has passed the validation set

70 – 80 % **10 – 15 %** **10 – 15 %**

jimutbahanpal@yahoo.com

VALIDATION

WORK FLOW



REPRESENTATION

Feature Engineering

Traditional programming focuses on code, but ML engineers focus on representation. The way developers hone a model is by adding and representing features. When we work with ML, the features doesn’t come as feature vectors in formatted options, instead as database records or protocol buffers.

REPRESENTATION

Feature Engineering

Traditional programming focuses on code, but ML engineers focus on representation. The way developers hone a model is by adding and representing features. When we work with ML, the features doesn't come as feature vectors in formatted options, instead as database records or protocol buffers.

Role of a ML engineer

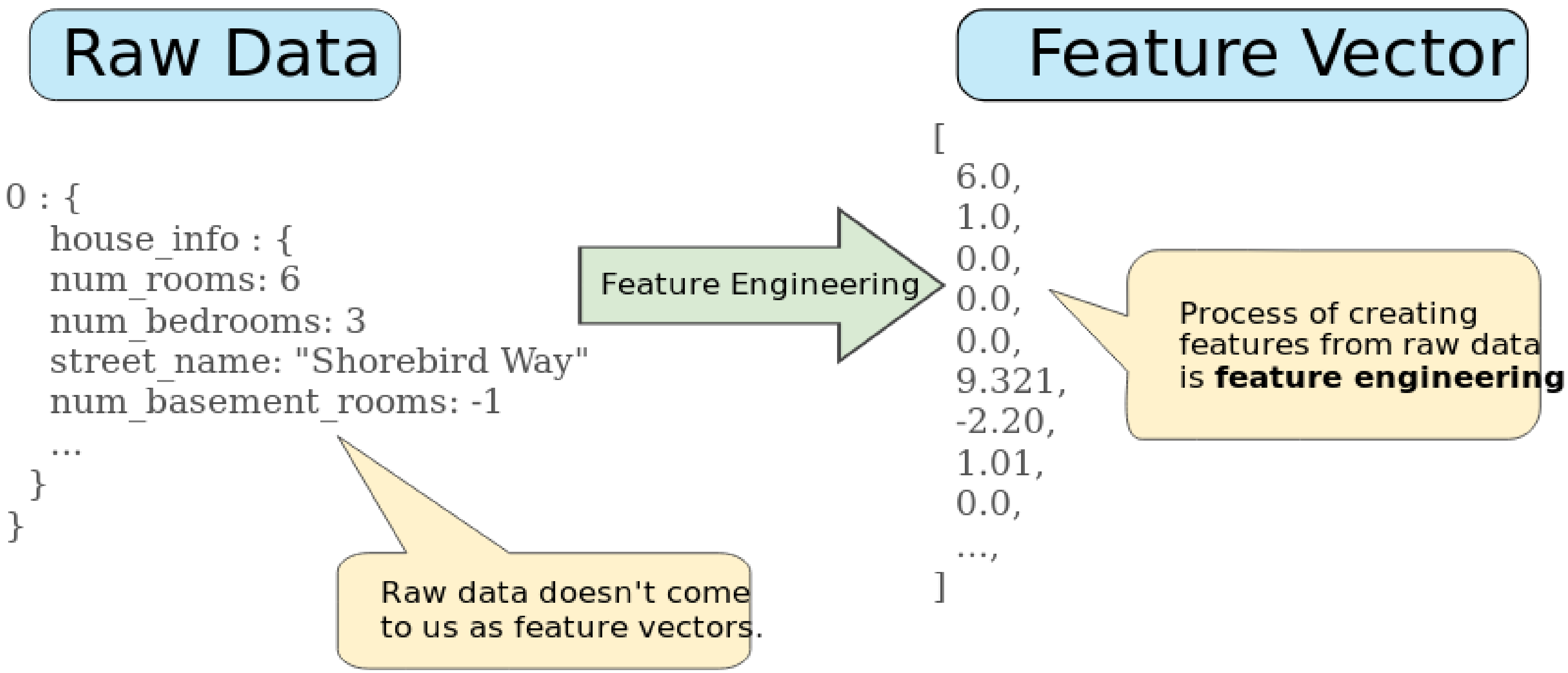
The ML engineer takes data from heterogenous sources and create feature vectors from it. The process of extracting features from raw data is called as feature engineering. This is about 75% of the time spent by a ML scientist/Engineer.

REPRESENTATION

Many machine learning models must represent the features as real-numbered vectors since the feature values must be multiplied by the model weights.

REPRESENTATION

Many machine learning models must represent the features as real-numbered vectors since the feature values must be multiplied by the model weights.



Feature engineering maps raw data to ML features.

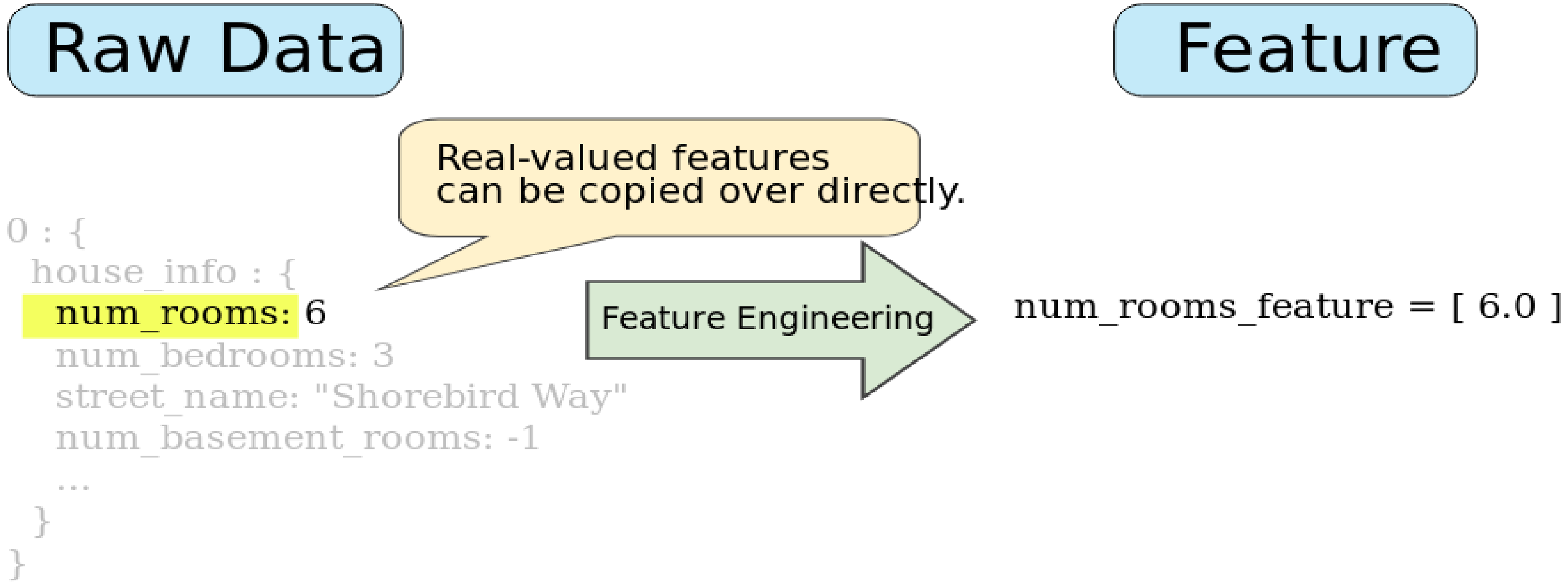
Mapping numeric values

Integer and floating-point data don't need a special encoding because they can be multiplied by a numeric weight. So, converting the raw integer value 6 to the feature value 6.0 is trivial

REPRESENTATION

Mapping numeric values

Integer and floating-point data don't need a special encoding because they can be multiplied by a numeric weight. So, converting the raw integer value 6 to the feature value 6.0 is trivial



Mapping integer values to floating-point values.

REPRESENTATION

Mapping categorical values – One Hot Encoding

Categorical features have a discrete set of possible values. For example, there might be a feature called street_name with options that include

REPRESENTATION

Mapping categorical values – One Hot Encoding

Categorical features have a discrete set of possible values. For example, there might be a feature called street_name with options that include

{'Charleston Road', 'North Shoreline Boulevard', 'Shorebird Way', 'Rengstorff Avenue'}

REPRESENTATION

Mapping categorical values – One Hot Encoding

Categorical features have a discrete set of possible values. For example, there might be a feature called street_name with options that include

{'Charleston Road', 'North Shoreline Boulevard', 'Shorebird Way', 'Rengstorff Avenue'}

We can instead create a binary vector for each categorical feature in our model that represents values as follows

jimutbahanpal@yahoo.com

REPRESENTATION

Mapping categorical values – One Hot Encoding

Categorical features have a discrete set of possible values. For example, there might be a feature called street_name with options that include

{'Charleston Road', 'North Shoreline Boulevard', 'Shorebird Way', 'Rengstorff Avenue'}

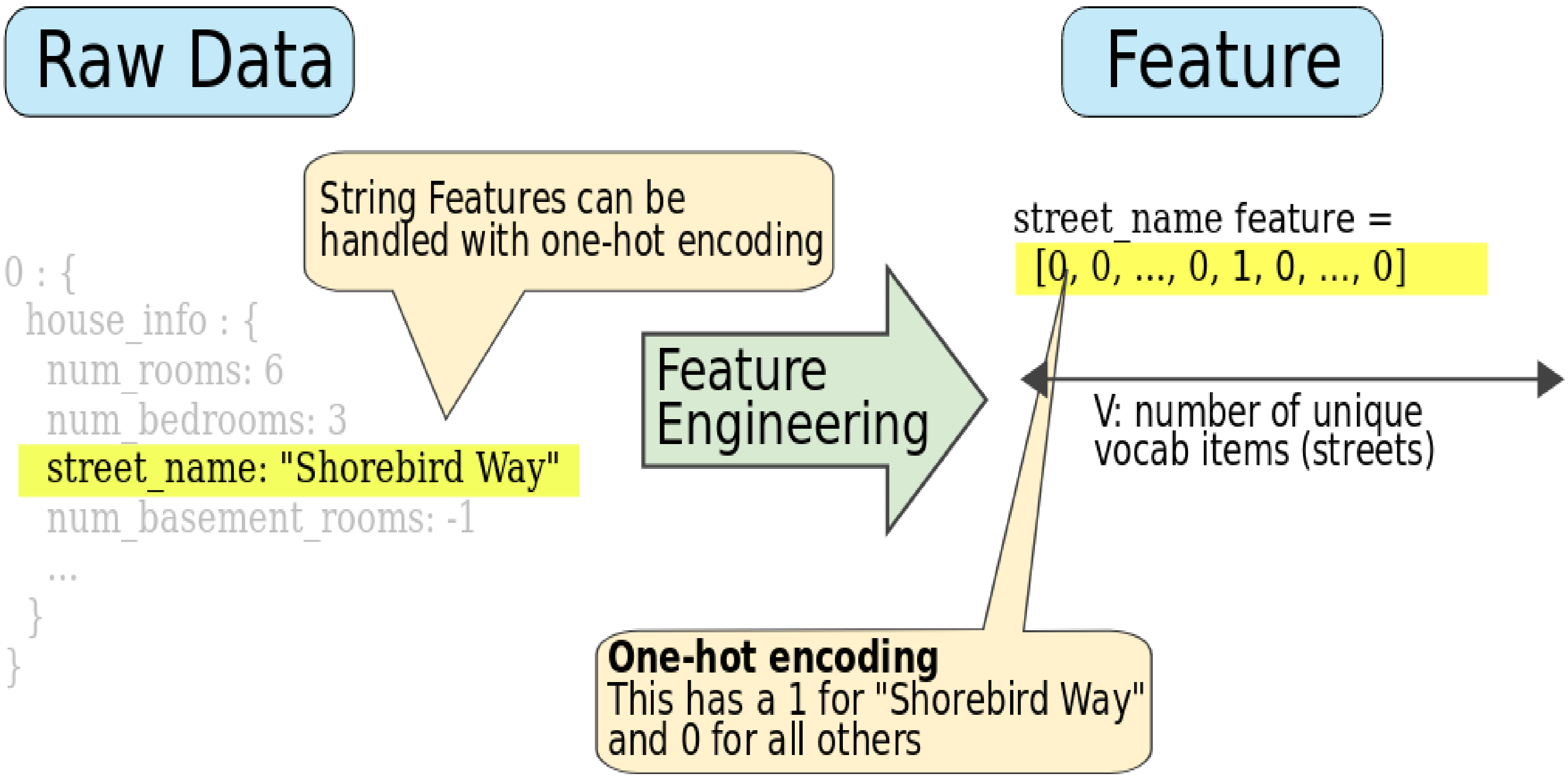
We can instead create a binary vector for each categorical feature in our model that represents values as follows

- **For values that apply to the example, set corresponding vector elements to 1.**

- **Set all other elements to 0.**

REPRESENTATION

Mapping ‘street’ via One hot encoding



jimutbahanpal@yahoo.com

REPRESENTATION

Qualities of Good Features

- **Avoid rarely used discrete feature values : the feature should be non-zero atleast a handful of time in our dataset. Doing so enables a model to learn how this feature value relates to the label. That is, having many examples with the same discrete value gives the model a chance to see the feature in different settings, and in turn, determine when it's a good predictor for the label.**

jimutbahanpal@yahoo.com

REPRESENTATION

Qualities of Good Features

- **Avoid rarely used discrete feature values : the feature should be non-zero atleast a handful of time in our dataset. Doing so enables a model to learn how this feature value relates to the label. That is, having many examples with the same discrete value gives the model a chance to see the feature in different settings, and in turn, determine when it's a good predictor for the label.**

- **Prefer clear and obvious meanings : the values of the feature vector should be clear (not noisy), obvious and concise. The values should be checked whether it is correct when coming from the source. Eg. values of house_age should be in years not in month.**

jimutbahanpal@yahoo.com

REPRESENTATION

Qualities of Good Features

- **Avoid rarely used discrete feature values : the feature should be non-zero atleast a handful of time in our dataset. Doing so enables a model to learn how this feature value relates to the label. That is, having many examples with the same discrete value gives the model a chance to see the feature in different settings, and in turn, determine when it's a good predictor for the label.**

- **Prefer clear and obvious meanings : the values of the feature vector should be clear (not noisy), obvious and concise. The values should be checked whether it is correct when coming from the source. Eg. values of house_age should be in years not in month.**

- **Account for upstream instability : the definition of a feature shouldn't change over time.**

jimutbahanpal@yahoo.com

Cleaning Data

As an ML engineer, you'll spend enormous amounts of your time tossing out bad examples and cleaning up the salvageable ones

Cleaning Data

As an ML engineer, you'll spend enormous amounts of your time tossing out bad examples and cleaning up the salvageable ones

- Scaling feature values : Scaling means converting floating-point feature values from their natural range (for example, 100 to 900) into a standard range (for example, 0 to 1 or -1 to +1). If a feature set consists of only a single feature, then scaling provides little to no practical benefit. If, however, a feature set consists of multiple features, then feature scaling provides the following benefits :

REPRESENTATION

Cleaning Data

As an ML engineer, you'll spend enormous amounts of your time tossing out bad examples and cleaning up the salvageable ones

- Scaling feature values : Scaling means converting floating-point feature values from their natural range (for example, 100 to 900) into a standard range (for example, 0 to 1 or -1 to +1). If a feature set consists of only a single feature, then scaling provides little to no practical benefit. If, however, a feature set consists of multiple features, then feature scaling provides the following benefits :

- **Helps gradient descent converge more quickly.**

Cleaning Data

As an ML engineer, you'll spend enormous amounts of your time **tossing out bad examples** and cleaning up the salvageable ones

- **Scaling feature values** : Scaling means converting floating-point feature values from their natural range (for example, 100 to 900) into a standard range (for example, 0 to 1 or -1 to +1). If a feature set consists of only a single feature, then scaling provides little to no practical benefit. If, however, a feature set consists of multiple features, then feature scaling provides the following benefits :
    - Helps gradient descent converge more quickly.
    - Helps the model learn appropriate weights for each feature. Without feature scaling, the model will pay too much attention To the features having a wider range.

REPRESENTATION

Cleaning Data

As an ML engineer, you'll spend enormous amounts of your time tossing out bad examples and cleaning up the salvageable ones

- Scaling feature values : Scaling means converting floating-point feature values from their natural range (for example, 100 to 900) into a standard range (for example, 0 to 1 or -1 to +1). If a feature set consists of only a single feature, then scaling provides little to no practical benefit. If, however, a feature set consists of multiple features, then feature scaling provides the following benefits :

- **Helps gradient descent converge more quickly.**
- **Helps the model learn appropriate weights for each feature. Without feature scaling, the model will pay too much attention To the features having a wider range.**
- **Avoids the NaN trap in which one number exceeds the floating point precision limit during training, so other numbers on multiplication too becomes NaN.**

jimutbahanpal@yahoo.com

Cleaning Data

- Handling extreme outliers : Suppose there is a dataset for values of rooms per houses, and in that cluster, some celebrity's house's data is also included. So on an average if the no. of rooms be 8-10 the celebrity may have 30-40, now that will create a problem and be outlier when we plot a nox plot. It is better to remove those values from the cluster since the model will learn to predict abnormal values for normal people.

jimutbahanpal@yahoo.com

REPRESENTATION

Cleaning Data

- Handling extreme outliers : Suppose there is a dataset for values of rooms per houses, and in that cluster, some celebrity’s house’s data is also included. So on an average if the no. of rooms be 8-10 the celebrity may have 30-40, now that will create a problem and be outlier when we plot a nox plot. It is better to remove those values from the cluster since the model will learn to predict abnormal values for normal people.

- Scrubbing : It refers to the removal of :

  Omitted values: For instance, a person forgot to enter a value for a house's age.

REPRESENTATION

Cleaning Data

- **Handling extreme outliers** : Suppose there is a dataset for values of rooms per houses, and in that cluster, some celebrity’s house’s data is also included. So on an average if the no. of rooms be 8-10 the celebrity may have 30-40, now that will create a problem and be outlier when we plot a nox plot. It is better to remove those values from the cluster since the model will learn to predict abnormal values for normal people.

- Scrubbing : It refers to the removal of :
  - **Omitted values:** For instance, a person forgot to enter a value for a house's age.
  - **Duplicate examples:** For example, a server mistakenly uploaded the same logs twice.

REPRESENTATION

Cleaning Data

- **Handling extreme outliers** : Suppose there is a dataset for values of rooms per houses, and in that cluster, some celebrity's house's data is also included. So on an average if the no. of rooms be 8-10 the celebrity may have 30-40, now that will create a problem and be outlier when we plot a nox plot. It is better to remove those values from the cluster since the model will learn to predict abnormal values for normal people.

- **Scrubbing** : It refers to the removal of :

  Omitted values: For instance, a person forgot to enter a value for a house's age.

  Duplicate examples: For example, a server mistakenly uploaded the same logs twice.

  Bad labels: For instance, a person mislabeled a picture of an oak tree as a maple.

jimutbahanpal@yahoo.com

REPRESENTATION

MLCC_SXC

Cleaning Data

- **Handling extreme outliers** : Suppose there is a dataset for values of rooms per houses, and in that cluster, some celebrity's house's data is also included. So on an average if the no. of rooms be 8-10 the celebrity may have 30-40, now that will create a problem and be outlier when we plot a nox plot. It is better to remove those values from the cluster since the model will learn to predict abnormal values for normal people.

- Scrubbing : It refers to the removal of :

  Omitted values: For instance, a person forgot to enter a value for a house's age.

  Duplicate examples: For example, a server mistakenly uploaded the same logs twice.

  Bad labels: For instance, a person mislabeled a picture of an oak tree as a maple.

  Bad feature values: For example, someone typed in an extra digit, or a thermometer was left out in the sun.

jimutbahanpal@yahoo.com

MLCC_SXC

MACHINE LEARNING

CRASH COURSE

FEATURE CROSSES AND REGULARIZATION

Week 2

- Jimut Bahan Pal
jimutbahanpal@yahoo.com